

SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning

Long Chen
Zhejiang University
longc@zju.edu.cn

Hanwang Zhang
National University of Singapore
hanwangzhang@gmail.com

Jun Xiao
Zhejiang University
junx@cs.zju.edu.cn

Liqiang Nie
Shandong University
nieliqiang@gmail.com

Jian Shao
Zhejiang University
jshao@zju.edu.cn

Tat-Seng Chua
National University of Singapore
dcscts@nus.edu.sg

Abstract

Visual attention has been successfully applied in structural prediction tasks such as visual captioning and question answering. Existing visual attention models are generally spatial, i.e., the attention is modeled as spatial probabilities that re-weight the last conv-layer feature map of a CNN which encodes an input image. However, we argue that such spatial attention does not necessarily conform to the attention mechanism — a dynamic feature extractor that combines contextual fixations over time, as CNN features are naturally spatial, channel-wise and multi-layer. In this paper, we introduce a novel convolutional neural network dubbed SCA-CNN that incorporates Spatial and Channel-wise Attentions in a CNN. In the task of image captioning, SCA-CNN dynamically modulates the sentence generation context in multi-layer feature maps, encoding where (i.e., attentive spatial locations at multiple layers) and what (i.e., attentive channels) the visual attention is. We evaluate the SCA-CNN architecture on three benchmark image captioning datasets: Flickr8K, Flickr30K, and MSCOCO. SCA-CNN achieves significant improvements over state-of-the-art visual attention-based image captioning methods.

1. Introduction

Visual attention has been shown effective in various structural prediction tasks such as image/video captioning [29, 31] and visual question answering [4, 30, 28]. Its success is mainly due to the reasonable assumption that human vision does not tend to process a whole image in its entirety at once; instead, one only focuses on selective parts of the whole visual space when and where as needed [5].

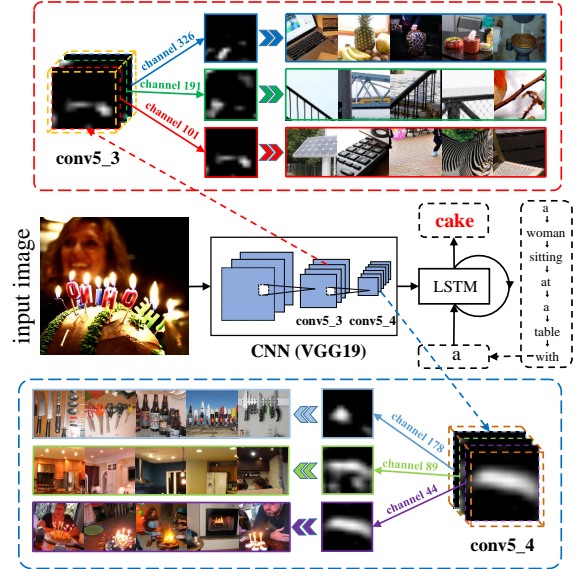


Figure 1. The illustration of channel-wise visual attention in two convolutional layers (*conv5_3* and *conv5_4* in VGG19) when predicting cake from the captioning a woman sitting at a table with cake. At each layer, top 3 attentive channels are visualized by showing the 5 most responsive receptive fields in the corresponding feature maps [35].

Specifically, rather than encoding an image into a static feature vector, attention allows the image feature to evolve from the sentence context at hand, resulting in richer and longer descriptions for cluttered images. In this way, visual attention can be considered as a dynamic feature extraction mechanism that combines contextual fixations over time [18, 23].

State-of-the-art image features are generally extracted by

deep Convolutional Neural Networks (CNNs) [8, 22]. Starting from an input color image of the size $W \times H \times 3$, a convolutional layer consisting of C -channel filters scans the input image and output a $W' \times H' \times C$ feature map, which will be the input for the next convolutional layer¹. Each 2D slice of a 3D feature map encodes the spatial visual responses raised by a filter channel, where the filter performs as a pattern detector — lower-layer filters detect low-level visual cues like edges and corners while higher-level ones detect high-level semantic patterns like parts and object [35]. By stacking the layers, a CNN extracts image features through a hierarchy of visual abstractions. Therefore, CNN image features are essentially *spatial*, *channel-wise*, and *multi-layer*. However, most existing attention-based image captioning models only take into account the spatial characteristic [29], *i.e.*, the attention models merely modulate the sentence context into the last conv-layer feature map via spatially attentive weights.

In this paper, we are going to take into full account of the three characteristics of CNN features for visual attention-based image captioning. In particular, we propose a novel Spatial and Channel-wise Attention-based Convolutional Neural Network, dubbed SCA-CNN, which learns to pay attention to every feature entry in the multi-layer 3D feature maps. Figure 1 illustrates the motivation of introducing channel-wise attention in multi-layer feature maps. First, since a channel-wise feature map is essentially a detector response map of the corresponding filter, channel-wise attention can be viewed as the process of selecting semantic attributes on the demand of the sentence context. For example, when we are going to predict *cake*, our channel-wise attention (*e.g.*, in the *conv5_3/conv5_4* feature map) will assign more weights on channel-wise feature maps generated by filters according to the semantics like *cake*, *fire*, *light*, and *candle-like* shapes. Second, as a feature map is dependent on its lower-layer ones, it is natural to apply the attention model in multiple layers, so as to gain visual attention on multiple semantic abstractions. For example, it is beneficial to emphasize on lower-layer channels corresponding to more elemental shapes like *array* and *cylinder* that compose *cake*.

We validate the effectiveness of the proposed SCA-CNN on three well-known image captioning benchmarks: Flickr8K, Flickr30K, and MSCOCO. We can significantly surpass the spatial attention model [29] by 4.8% in BLEU4. In summary, we propose a unified SCA-CNN framework to effectively integrate spatial, channel-wise, and multi-layer visual attention in CNN features for image captioning. In particular, a novel spatial and channel-wise attention model is proposed. This model is generic and thus can be applied to any layer in any CNN architecture such as the popular

VGG [22] and ResNet [8]. SCA-CNN helps us to gain a better understanding of how CNN features evolve in the process of sentence generation.

2. Related Work

We are interested in the visual attention models used in the encoder-decoder framework for neural image/video captioning (NIC) and visual question answering (VQA), which fall in the recent trend of connecting computer vision and natural language [13]. Pioneering works on NIC [27, 12, 6, 26, 25] and VQA [1, 16, 7, 20] use CNN to encode an image or video into a static visual feature vector and then feed it into an RNN [9] to decode the language sequences such as captions or answers.

However, the static vector does not allow the image feature adapting to the sentence context at hand. Inspired by the attention mechanism introduced in machine translation [2], where the decoder dynamically selects useful source language words or sub-sequence for the translation into target language, visual attention models had been widely-used in NIC and VQA. We categorize these attention-based models into the following three domains that motivate our SCA-CNN:

Spatial Attention. Xu *et al.* [29] proposed the first visual attention model in image captioning. In general, they used “hard” pooling that selects the most probably attentive region, or “soft” pooling that averages the spatial features with attentive weights. For example, in VQA, Zhu *et al.* [36] adopted the “soft” attention to select image regions while encoding question sentence. To further refine the spatial attention, Yang *et al.* [30] and Xu *et al.* [28] applied a stacked attention model, where the second attention is based on the attentive feature map modulated by the first one. Different from theirs, our multi-layer attention is applied on the multiple layers of a CNN. A common defect of the above spatial models is that they generally resort to weighted pooling on the attentive feature map. Thus, spatial information will be lost inevitably. More seriously, their attention is only applied in the last conv-layer, where the receptive field of a region will be quite large and the difference between the visual features of different regions is limited, resulting in insignificant spatial attentions.

Semantic Attention. Besides the spatial information, You *et al.* [32] proposed to select semantic concepts in NIC, where the image feature is a vector of confidences of attribute classifiers. Jia *et al.* [11] exploited the correlation between images and their captions as the global semantic information to guide the LSTM generating sentences. However, these models require external resources to train the semantic attributes. In SCA-CNN, we do not need such data as the filters of a CNN can be considered as semantic detectors [35]. Therefore, the channel-wise attention of SCA-CNN is similar to semantic attention.

¹Each convolutional layer is optionally followed by a pooling, down-sampling, normalization, or a fully connected layer.

Multi-layer Attention. According to the nature of CNN architecture, the sizes of respective fields corresponding to different feature map layers are different. To overcome the weakness of large respective field size in the last conv-layer attention, Seo *et al.* [21] proposed a multi-layer attention networks [21]. As compared to theirs, SCA-CNN also incorporates the channel-wise attention at multiple layers.

3. Spatial and Channel-wise Attention CNN

3.1. Overview

We adopt the popular encoder-decoder framework for image caption generation, where a CNN first encodes an input image into a fixed-length vector and then an LSTM decodes the vector into a sequence of words. As illustrated in Figure 2, SCA-CNN endows the original CNN multi-layer feature maps adaptive to the sentence context through channel-wise attention and spatial attention at multiple layers.

Formally, suppose we are going to generate the t -th word of the image caption. At hand, we have the last sentence context encoded in the LSTM memory $\mathbf{h}_{t-1} \in \mathbb{R}^d$, where d is the embedding dimension. At the l -th layer, the memory \mathbf{h}_{t-1} is going to determine the spatial attention α^l , the channel-wise attention β^l , and the feature map \mathbf{X}^l modulated by the attentions:

$$\begin{aligned} \{\alpha^l, \beta^l\} &= \Phi(\mathbf{h}_{t-1}, \mathbf{V}^l), \\ \mathbf{X}^l &= f(\mathbf{V}^l, \alpha^l, \beta^l). \end{aligned} \quad (1)$$

where $\Phi()$ is the spatial and channel-wise attention function that will be detailed in Section 3.2 and 3.3; $f()$ is the modulate function that does linear combination between input feature maps and attention weights; $\mathbf{X}^l \in \mathbb{R}^{W^l \times H^l \times C^l}$ is the modulated feature map of the size $W^l \times H^l \times C^l$ at the l -th layer. Note that \mathbf{V}^l is the feature map output from the previous conv-layer, *e.g.*, convolution followed by pooling or down-sampling [22, 8]:

$$\mathbf{V}^l = \text{CNN}(\mathbf{X}^{l-1}). \quad (2)$$

So far, we are ready to generate the t -th word by:

$$\begin{aligned} \mathbf{h}_t &= \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{X}^L, y_{t-1}), \\ y_t \sim p_t &= \text{softmax}(\mathbf{h}_t, y_{t-1}). \end{aligned} \quad (3)$$

where L is the total number of conv-layers; $p_t \in \mathbb{R}^{|\mathcal{D}|}$ is a probability vector and \mathcal{D} is a predefined dictionary.

Let's first separate the attention function $\Phi()$ in Eq (1) into two parts: the spatial attention part Φ_s and channel-wise attention part Φ_c . That is,

$$\alpha^l = \Phi_s(\mathbf{h}_{t-1}, \mathbf{V}^l), \quad (4)$$

$$\beta^l = \Phi_c(\mathbf{h}_{t-1}, \mathbf{V}^l). \quad (5)$$

3.2. Spatial Attention

In general, a word only relates to a small part of an image. For example, in Figure 1, when we are going to predict cake, only image regions which contain cake are useful. Therefore, applying a global image feature vector to generate captions may lead to sub-optimal results due to the irrelevant regions. Instead of considering each image region equally, spatial attention mechanism attempts to pay more attention to the semantic-related regions. Without loss of generality, we omit the superscript l for the l -th layer. We rewrite $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$, $\mathbf{V} \in \mathbb{R}^{C \times m}$, where m is the number of image regions, C is the number of channels. So $\mathbf{v}_i \in \mathbb{R}^C$ is visual feature in the i -th location. Given the previous time step LSTM hidden state \mathbf{h}_{t-1} , we use a single-layer neural network followed by a softmax function to generate the attention distributions α over the image regions. Here are the definition of spatial attention model Φ_s :

$$\begin{aligned} \mathbf{a} &= \tanh((\mathbf{W}_s \mathbf{V} + b_s) \oplus \mathbf{W}_{hs} \mathbf{h}_{t-1}), \\ \alpha &= \text{softmax}(\mathbf{W}_i \mathbf{a} + b_i). \end{aligned} \quad (6)$$

where $\mathbf{W}_s \in \mathbb{R}^{k \times C}$, $\mathbf{W}_{hs} \in \mathbb{R}^{k \times d}$ are transformation matrix that mapping image visual feature and hidden state to a same dimension. We denote \oplus as the addition of a matrix and a vector. The addition between a matrix and a vector is performed by adding each column of the matrix by the vector. $b_s \in \mathbb{R}^k$, $b_i \in \mathbb{R}^1$ are bias terms for linear transformation.

There exists many strategies to obtain a weighted visual feature based on attention probability. The most common one is summing up the weighted feature to generate a single vector representation [29]. In contrast, SCA-CNN feeds the weighted visual feature to the next layer for further processing.

3.3. Channel-wise Attention

Note that the spatial attention function in Eq (4) still requires the visual features \mathbf{V} to calculate attention, but the feature used in spatial attention is in fact not attention-based. So, we introduce a channel-wise attention mechanism to attend features \mathbf{V} . It is worth noting that CNN filters perform as a pattern detector, and each channel-wise feature map is the response map of the corresponding filter. Therefore, channel-wise attention can be viewed as the process of selecting semantic attributes.

For channel-wise attention, we first apply spatial mean pooling for each channel to obtain the channel feature \mathbf{v} :

$$\mathbf{v} = [v_1, v_2, \dots, v_C], \mathbf{v} \in \mathbb{R}^C \quad (7)$$

where C is the number of channels, scalar v_i represents i -th channel feature. Following the recipe of spatial attention, we can obtain the channel-wise attention weights β . Here

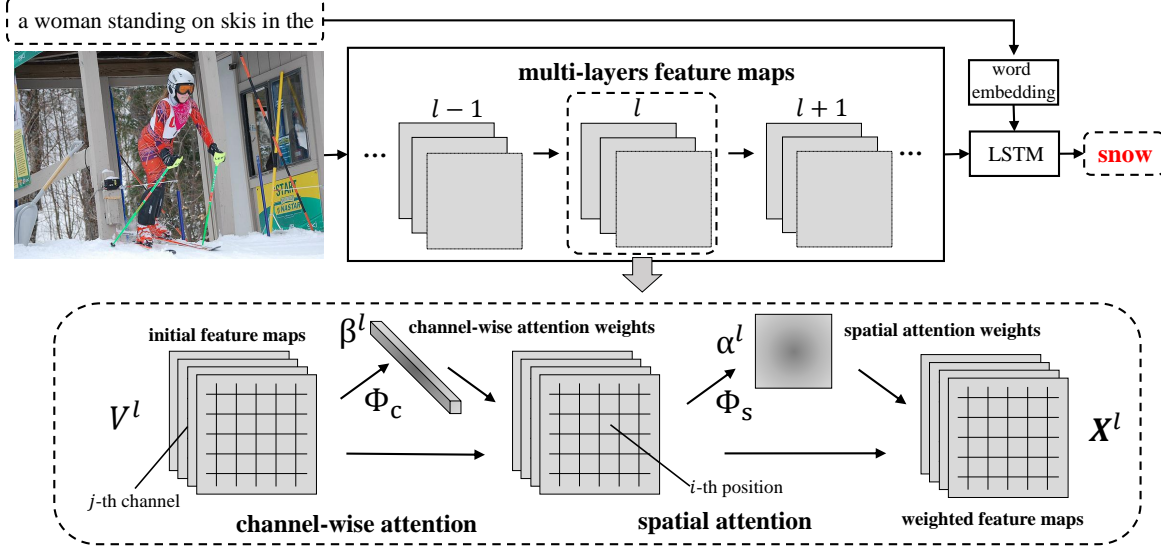


Figure 2. The overview of our proposed SCA-CNN framework. For the l -th layer, the initial feature map \mathbf{V}^l is the output of the $(l-1)$ -th layer. We first use the channel-wise attention function Φ_c to obtain the channel-wise attention weight β^l , which is multiplied in every channel of the feature map. Then, we use the spatial-wise attention function Φ_s to obtain the spatial attention weight α^l , which is multiplied in every spatial regions, resulting in the final attentive feature map \mathbf{X}^l . The order of spatial and channel-wise attention is discussed in Section 3.3.

are the definition of channel-wise attention model Φ_c :

$$\begin{aligned} \mathbf{b} &= \tanh((\mathbf{W}_c \otimes \mathbf{v} + b_c) \oplus \mathbf{W}_{hc} \mathbf{h}_{t-1}), \\ \beta &= \text{softmax}(\mathbf{W}'_i \mathbf{b} + b'_i). \end{aligned} \quad (8)$$

where $\mathbf{W}_c \in \mathbb{R}^k$, $\mathbf{W}_{hc} \in \mathbb{R}^{k \times d}$ are transformation matrix, \otimes represents the outer product of vectors. $b_c \in \mathbb{R}^k$, $b'_i \in \mathbb{R}^1$ are bias terms.

According to different implementation order of channel-wise attention and spatial attention, there exists three types of model which incorporating channel-wise attention and spatial attention:

Channel-Spatial. The first type is channel-wise attention followed by spatial attention as illustrated in Figure. 2. At first, for each initial feature map \mathbf{V} , we adopt channel-wise attention Φ_c to obtain the channel-wise attention weights β . Through linear combination of β and \mathbf{V} , we get a channel-wise weighted feature map. Then we feed the channel-wise weighted feature map to spatial attention model Φ_s and obtain the spatial weights α . After getting two attention weights α and β , we can feed \mathbf{V} , β , α to modulate function f to calculate the modulated feature map \mathbf{X} . All the processes are summarized as follows:

$$\begin{aligned} \beta &= \Phi_c(\mathbf{h}_{t-1}, \mathbf{V}), \\ \alpha &= \Phi_s(\mathbf{h}_{t-1}, f_c(\mathbf{V}, \beta)), \\ \mathbf{X} &= f(\mathbf{V}, \alpha, \beta) \end{aligned} \quad (9)$$

where $f_c()$ is a multiplication for feature map channels and corresponding channel weights.

Spatial-Channel. The second type is channel-wise attention followed by spatial attention. For each initial feature map \mathbf{V} , we first utilize spatial attention Φ_s to obtain the spatial attention weights α . Based on α and Φ_c , we can get β and \mathbf{X} :

$$\begin{aligned} \alpha &= \Phi_s(\mathbf{h}_{t-1}, \mathbf{V}), \\ \beta &= \Phi_c(\mathbf{h}_{t-1}, f_s(\mathbf{V}, \alpha)), \\ \mathbf{X} &= f(\mathbf{V}, \alpha, \beta) \end{aligned} \quad (10)$$

where $f_s()$ is a multiplication for feature map regions and corresponding region weights.

Spatial and Channel Unlike the above two types of attention models which consider spatial attention and channel-wise attention in two separate steps, the third type of model integrate them into one step. For a feature map \mathbf{V} , it directly use attention model Φ to get the attention weights γ . Then linear combine γ and \mathbf{V} we can directly get \mathbf{X} :

$$\begin{aligned} \gamma &= \Phi(\mathbf{h}_{t-1}, \mathbf{V}), \\ \mathbf{X} &= f(\mathbf{V}, \gamma). \end{aligned} \quad (11)$$

In this model, each scalar feature has a probability weight, which can significantly increase the variations of feature representation. Unfortunately, it needs far more computing resources and make it unrealistic for local experiments.

4. Experiments

We are going to validate the effectiveness of the proposed SCA-CNN for image captioning by answering the

following questions:

- Q1** Is the channel-wise attention effective? Will it improve the spatial attention?
- Q2** Is the multi-layer attention effective?
- Q3** How does SCA-CNN perform compared to other state-of-the-art visual attention models?

4.1. Dataset and Metric

We conducted experiments on three well-known benchmarks: 1) **Flickr8k** [10]: it contains 8,000 images. Following the official split, it selects 6,000 images for training, 1,000 images for validation, and 1,000 images for testing; 2) **Flickr30k** [33]: it contains 31,000 images. Because of the lack of official split, for fair comparison with previous work, we report with the publicly available split used in previous work [12]. In this split, 29,000 images are used for training, 1,000 images are used for validation, and 1,000 images are used for testing; and 3) **MS COCO** [15]: it contains 82,783 images as training set, 40,504 images as validation set and 40,775 images as test set. As the ground truth of the MS COCO test set is unknown, the validation set is further split into validation subset for model selection and testing subset for local experiments. The split is also following [12]. It uses the whole training set 82,783 images for training, respectively select 5,000 images from validation set for validation and 5,000 images from validation set for test. The Flickr8k and Flickr30k datasets are both annotated with 5 sentences per image, but for MSCOCO, some of the images have more than 5 sentences. As for the sentences preprocessing, we followed the publicly available code¹ to do some basic preprocess (*i.e.* building dictionaries, tokenizing the captions, convert word to lowercase). We used BLEU (**B@1**, **B@2**, **B@3**, **B@4**) [19], METEOR (**MT**) [3], CIDEr(**CD**) [24], and ROUGE-L (**RG**) [14] as evaluation metrics. In a nutshell, they measure the consistency between n-gram occurrences in generated and ground-truth sentences, where this consistency is weighted by n-gram saliency and rarity. We used the Microsoft COCO caption evaluation tool² for implementation of the four metrics.

4.2. Setup

Our captioning system is implemented based on the widely-used encoder-decoder pipeline. For the encoder side, we used the proposed SCA-CNN to encode a input image. We adopted two widely-used architectures: VGG-19 [22] and ResNet-152 [8] as the base CNNs for SCA-CNN. For the decoder side, we used the LSTM implementation [9]. The word embedding and hidden dimension were respectively set to 100 and 1000. The common

¹<https://github.com/karpathy/neuraltalk>

²<https://github.com/tylin/coco-caption>

space dimension for calculating attention weights was set to 512. And weight decay coefficient was set to $1e-4$. For Flickr8k, we set the minibatch size to 16, and for Flickr30k and MS COCO, we set the number to 64. For Flickr8k, we utilized all words in training set to create the dictionary, and for Flickr30k and MS COCO, we ignored the words appearing less than 5 times (nearly 6000 words for Flickr8k, nearly 10000 words for Flickr30k and MS COCO). We used dropout and early stopping to avoid overfitting and used validation set log-likelihood for model selection. Our whole framework was trained in an end-to-end way with Adadelta [34], which is a stochastic gradient descent method using an adaptive learning rate algorithm. In testing, a caption was formed by drawing words from RNN until a special end token is reached. We followed the strategy of BeamSearch [27] to generate a sentence given an image, and the beam size was set to 5. We noticed that the trick of the beam search with length normalization [11] can also help to improve performance at some degree. But for fair comparison, all results reported are without length normalization.

4.3. Evaluations of Channel-wise Attention (Q1)

Comparing Methods. We first compare the spatial attention and channel-wise attention methods. 1) **S**: It is a spatial attention model. Obtaining the spatial attention weights based on the last conv-layer, the weighted spatial feature are fed into the following conv-layer. For VGG-19, we used *conv5_4* layer feature map and followed by two full connected layers. For ResNet-152, we used *res5c* layer feature map and followed by an average pooling. 2) **C**: This is a channel level attention model in Eq. (5). This model is only based on channel-wise attention without any spatial attention. 3) **S-C**: Our spatial and channel-wise model incorporating two types attention in Eq. (10), with spatial attention implemented first. 4) **C-S**: Our channel-wise and spatial model in Eq. (9) with channel-wise attention implemented first. 5) **SAT** [29]: We used hard attention model, which obtains better performance than the “soft” attention model among different datasets and metrics. As for the results showed in Table 1, the VGG results come from the original paper and the ResNet results come from our own implementation.

Results From Table 1, we have the following observations: 1) Using VGG-19, S is better than SAT; Using ResNet-152, SAT performance is better than S. This because VGG-19 network has fully-connected layers, which preserve the spatial information, however, ResNet-152 are originally followed by average pooling, which can not preserve the spatial information. 2) Comparing performance of C and S, ResNet-152 can improve C performance significantly better than VGG-19 network, which shows more channel numbers can improve the channel-wise attention

Dataset	Network	Method	B@4	MT	RG	CD
Flickr8k	VGG	S	23.0	21.0	49.1	60.6
		SAT	21.3	20.3	—	—
		C	22.6	20.3	48.7	58.7
		S-C	22.6	20.9	48.7	60.6
		C-S	23.5	21.1	49.2	60.3
	ResNet	S	20.5	19.6	47.4	49.9
		SAT	21.7	20.1	48.4	55.5
		C	24.4	21.5	50.0	65.5
		S-C	24.8	22.2	50.5	65.1
		C-S	25.7	22.1	50.9	66.5
Flickr30k	VGG	S	21.1	18.4	43.1	39.5
		SAT	19.9	18.5	—	—
		C	20.1	18.0	42.7	38.0
		S-C	20.8	17.8	42.9	38.2
		C-S	21.0	18.0	43.3	38.5
	ResNet	S	20.5	17.4	42.8	35.3
		SAT	20.1	17.8	42.9	36.3
		C	21.5	18.4	43.8	42.2
		S-C	21.9	18.5	44.0	43.1
		C-S	22.1	19.0	44.6	42.5
MS COCO	VGG	S	28.2	23.3	51.0	85.7
		SAT	25.0	23.0	—	—
		C	27.3	22.7	50.1	83.4
		S-C	28.0	23.0	50.6	84.9
		C-S	28.1	23.5	50.9	84.7
	ResNet	S	28.3	23.1	51.2	84.0
		SAT	28.4	23.2	51.2	84.9
		C	29.5	23.7	51.8	91.0
		S-C	29.8	23.9	52.0	91.2
		C-S	30.4	24.5	52.5	91.7

Table 1. The performance of spatial only (S), hard attention(SAT), channel only (C), spatial channel order (S-C), and channel spatial order (C-S) with one layer attention in VGG-19 Network and ResNet-152 Network.

performance as ResNet-152 has more channel numbers than VGG-19. 3) In both VGG-19 and ResNet-152, the performances of S-C and S-C are very similar. Generally, C-S is slightly better than S-C, which is perhaps due to that channel-wise features are more attentive. 4) In ResNet-152, C-S or S-C can obviously improve the performance of S. This demonstrates that by adding channel-wise attention, we can improve performance significantly when channels numbers are large.

4.4. Evaluations of Multi-layer Attention (Q2)

Comparing Methods We are going to investigate whether we can improve spatial attention or channel-wise attention by adding more attentive layers. In particular, we denote **1-layer**, **2-layer**, **3-layer** as the number of layers we used in spatial or spatial and channel-wise model. For VGG-19, the 1-st layer, 2-nd layer, 3-rd layer are respectively *conv5_4*, *conv5_3*, *conv5_2* conv-layer. For ResNet-

Dataset	Network	Method	B@4	MT	RG	CD
Flickr8k	VGG	1-layer	23.0	21.0	49.1	60.6
		2-layer	22.8	21.2	49.0	60.4
		3-layer	21.6	20.9	48.4	54.5
	ResNet	1-layer	20.5	19.6	47.4	49.9
		2-layer	22.9	21.2	48.8	58.8
		3-layer	23.9	21.3	49.7	61.7
Flickr30k	VGG	1-layer	21.1	18.4	43.1	39.5
		2-layer	21.9	18.5	44.3	39.5
		3-layer	20.8	18.0	43.0	38.5
	ResNet	1-layer	20.5	17.4	42.8	35.3
		2-layer	20.6	18.6	43.2	39.7
		3-layer	21.0	19.2	43.4	43.5
MS COCO	VGG	1-layer	28.2	23.3	51.0	85.7
		2-layer	29.0	23.6	51.4	87.4
		3-layer	27.4	22.9	50.4	80.8
	ResNet	1-layer	28.3	23.1	51.2	84.0
		2-layer	29.7	24.1	52.2	91.1
		3-layer	29.6	24.2	52.1	90.3

Table 2. The performance of multi-layer in spatial attention (S) in both VGG-19 network and ResNet-152 network

Dataset	Network	Method	B@4	MT	RG	CD
Flickr8k	VGG	1-layer	23.5	21.1	49.2	60.3
		2-layers	22.8	21.6	49.5	62.1
		3-layers	22.7	21.3	49.3	62.3
	ResNet	1-layer	25.7	22.1	50.9	66.5
		2-layers	25.8	22.4	51.3	67.1
		3-layers	25.3	22.9	51.2	67.5
Flickr30k	VGG	1-layer	21.0	18.0	43.3	38.5
		2-layers	21.8	18.8	43.7	41.4
		3-layers	20.7	18.3	43.6	39.2
	ResNet	1-layer	22.1	19.0	44.6	42.5
		2-layers	22.3	19.5	44.9	44.7
		3-layers	22.0	19.2	44.7	42.8
MS COCO	VGG	1-layer	28.1	23.5	50.9	84.7
		2-layers	29.8	24.2	51.9	89.7
		3-layers	29.4	24.0	51.7	88.4
	ResNet	1-layer	30.4	24.5	52.5	91.7
		2-layers	31.1	25.0	53.1	95.2
		3-layers	30.9	24.8	53.0	94.7

Table 3. The performance of multi-layer in combined attention layers (C-S) in both VGG-19 network and ResNet-152 network

152, it represents *res5c*, *res5c_branch2b*, *res5c_branch2a* conv-layer. Specifically, our strategy for training more attentive layers is to utilize the previous attention layers as initialization, which can significantly reduce the training time and achieve better results of randomly initialized training.

Results From Table 2 and 3, we have following observations: 1) In most of experiments, by adding layers in both two models (S and C-S) can achieve better results. This is because multi-layer attention can help to gain visual attention on multiple semantic abstractions. 2) Too many

Model	Flickr8k					Flickr30k					MS COCO				
	B@1	B@2	B@3	B@4	MT	B@1	B@2	B@3	B@4	MT	B@1	B@2	B@3	B@4	MT
Deep VS [12]	57.9	38.3	24.5	16.0	–	57.3	36.9	24.0	15.7	–	62.5	45.0	32.1	23.0	19.5
Google NIC [27] [†]	63.0	41.0	27.0	–	–	66.3	42.3	27.7	18.3	–	66.6	46.1	32.9	24.6	–
m-RNN [17]	–	–	–	–	–	60.0	41.0	28.0	19.0	–	67.0	49.0	35.0	25.0	–
Soft-Attention [29]	67.0	44.8	29.9	19.5	18.9	66.7	43.4	28.8	19.1	18.5	70.7	49.2	34.4	24.3	23.9
Hard-Attention [29]	67.0	45.7	31.4	21.3	20.3	66.9	43.9	29.6	19.9	18.5	71.8	50.4	35.7	25.0	23.0
emb-gLSTM [11]	64.7	45.9	31.8	21.2	20.6	64.6	44.6	30.5	20.6	17.9	67.0	49.1	35.8	26.4	22.7
ATT [32] [†]	–	–	–	–	–	64.7	46.0	32.4	23.0	18.9	70.9	53.7	40.2	30.4	24.3
SCA-VGG	65.5	46.6	32.6	22.8	21.6	64.6	45.3	31.7	21.8	18.8	70.5	53.3	39.7	29.8	24.2
SCA-RES	68.2	49.6	35.9	25.8	22.4	66.2	46.8	32.5	22.3	19.5	71.9	54.8	41.1	31.1	25.0

Table 4. Performances compared with the state-of-art in Flickr8k, Flickr30k and MS COCO dataset. SCA-VGG is our C-S 2-layer model based on VGG-19 network, and SCA-RES is our C-S 2-layer model based on ResNet-152 network. [†] indicates an ensemble model results. (–) indicates an unknown metric

Model	B@1		B@2		B@3		B@4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCA	71.2	89.4	54.2	80.2	40.4	69.1	30.2	57.9	24.4	33.1	52.4	67.4	91.2	92.1
Hard-Attention	70.5	88.1	52.8	77.9	38.3	65.8	27.7	53.7	24.1	32.2	51.6	65.4	86.5	89.3
ATT [†]	73.1	90.0	56.5	81.5	42.4	70.9	31.6	59.9	25.0	33.5	53.5	68.2	95.3	95.8
Google NIC [†]	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	53.0	68.2	94.3	94.6

Table 5. Performances of the proposed attention model on the online MS COCO testing server. [†] indicates an ensemble model results.

layers can also result in severe overfitting. For example, Flickr8k’s performance is more easier to degrade than MS COCO when adding more attention layers, as the size of Flickr8k is much smaller than MS COCO.

4.5. Comparison with State-of-The-Arts (Q3)

Comparing Methods We compared the proposed SCA-CNN with state-of-the-art methods for caption generation. 1) **Deep VS** [12] and **m-RNN** [17] are both end-to-end multimodal recurrent neural network for caption generation. 2) **Google NIC** [27]: A single network combine deep CNNs for image encoding and an LSTM for sequence modeling. 3) **Soft-Attention** [29]: A spatial attention model which use the weighted sum of image region features as the attend feature. 4) **Hard-Attention** [29]: A spatial attention model which apply random sampling on region features to represent visual information. 5) **emb-gLSTM** [11]: A semantic model adopt correlation between image and its description as the global semantic information. 6) **ATT** [32]: A semantic attention model which utilizing visual concepts corresponded word as semantic information. For both VGG-19 and ResNet-152 model, we only report results from 2-layer C-S model in this Table 4, since this model obtains the best performance among most of the different models in the previous experiments.

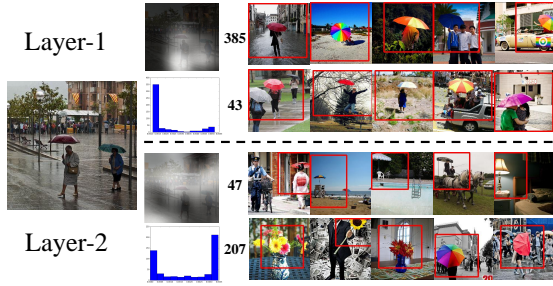
Besides the three benchmarks, we also evaluate our model on MS COCO Image Challenge set c5 and c40 by

uploading results to the official test sever and report the results in Table 5.

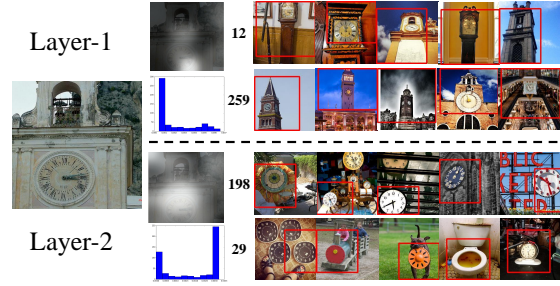
Results From Table 4 and Table 5, we can see that in most of the cases, SCA-CNN outperforms other models. This is due to that SCA-CNN exploits spatial, channel-wise, and multi-layer attentions, while most of the attention models only focus on one attention type. Note that the reason why we cannot surpass ATT and Google NIC is because they claim to use ensemble models. However, as a single model, SCA-CNN can still achieve comparative results compared to the ensemble models. In local experiments, for COCO dataset, ATT model performance surpass SCA-CNN only 0.6% in BLEU4 and 0.1% in METEOR. In COCO server, Google NIC surpass SCA-CNN only 0.7% in BLEU4 and 1% in METEOR.

4.6. Visualization of Spatial and Channel Attention

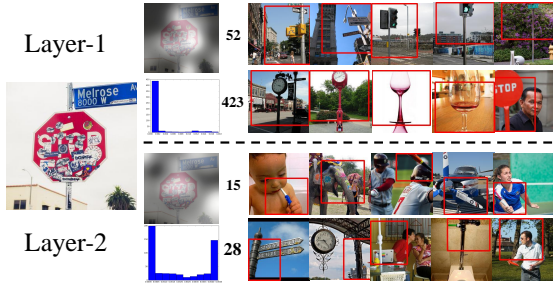
We provide some qualitative examples in Figure 3 for better understanding of our model. For simplicity, we only show the results at one word prediction step. For example in the first sample, when SCA-CNN model try to predict the *umbrella*, our channel-wise attention will assign more weights on channel-wise feature maps generated by filters according to the semantics like umbrella, stick, and round-like shape. The histogram in each layer indicates the probability distribution of all channels. And the visualization of attention map is the spatial attention map. For



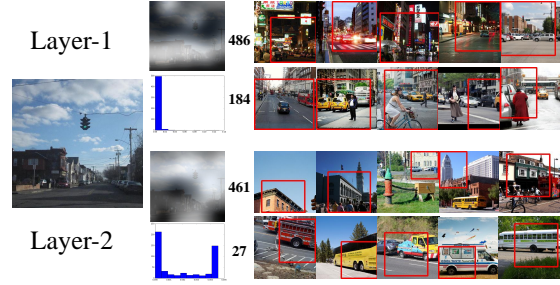
Ours: a woman walking down a street holding an **umbrella**
 SAT: a group of people standing next to each other
 GT: two females walking in the rain with umbrellas



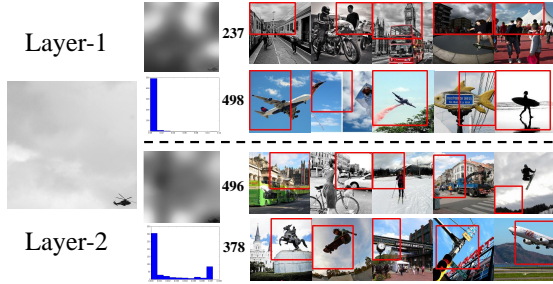
Ours: a **clock** tower in the middle of a city
 SAT: a clock tower on the side of a building
 GT: there is an old clock on top of a bell tower



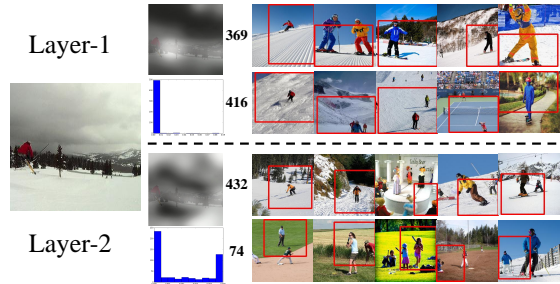
Ours: a street sign on a **pole** in front of a building
 SAT: a street sign in front of a building
 GT: a stop sign is covered with stickers and graffiti



Ours: a traffic light in the middle of a **city** street
 SAT: a group of people walking down a street
 GT: a street light at an intersection in a small town



Ours: a plane flying in the sky over a **cloudy** sky
 SAT: a plane flying through the sky in the sky
 GT: a couple of helicopters are in the sky



Ours: a man riding skis down a snow covered **slope**
 SAT: a man riding a snowboard down a snowy hill
 GT: a person riding skis goes down a snowy path

Figure 3. Examples of visualization results on spatial attention and channel-wise attention. Each examples contains three captions. Ours(SCA-CNN model), SAT(hard-attention model) and GT(ground truth). The number in the third column is the channel number of VGG-19 with highest channel attention wights, and next five images are the images with high activation in the corresponding channel number. The red box is respective field of corresponding layers

each layer we select two channels with highest probability. And We show 5 images from MS COCO train set with high activation in the corresponding channel to demonstrate the channel semantic information.

5. Conclusions

We proposed a novel attention architecture dubbed SCA-CNN for image captioning. SCA-CNN takes the full account of the characteristics of CNN into attentive image features: spatial, channel-wise, and multi-layer, achieving state-of-the-art performance across popular benchmarks.

The contribution of SCA-CNN is not only the more powerful attention model, but also a better understanding of where (*i.e.*, spatial) and what (*i.e.*, channel-wise) the attention looks like in a CNN that evolves during the sentence generation. Moving forward, we are going to bring temporal attention in SCA-CNN, so as to attend features in different video frames for video captioning. We are also going to investigate how to increase the number of attentive layers without severe overfitting.

References

- [1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 2
- [2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2014. 2
- [3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL*, 2005. 5
- [4] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering. In *CVPR*, 2016. 1
- [5] M. Corbetta and G. L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience*, 2002. 1
- [6] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2
- [7] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *NIPS*, 2015. 2
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016. 1, 2, 3, 5
- [9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 2, 5
- [10] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 2013. 5
- [11] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guiding the long-short term memory model for image caption generation. In *ICCV*, 2015. 2, 5, 7
- [12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 2, 5, 7
- [13] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowd-sourced dense image annotations. *IJCV*, 2016. 2
- [14] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL*, 2004. 5
- [15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5
- [16] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 2
- [17] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). In *ICLR*, 2015. 7
- [18] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NIPS*, 2014. 1
- [19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 5
- [20] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015. 2
- [21] P. H. Seo, Z. Lin, S. Cohen, X. Shen, and B. Han. Hierarchical attention networks. *arXiv preprint arXiv:1606.02393*, 2016. 3
- [22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2, 3, 5
- [23] M. F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep networks with internal selective attention through feedback connections. In *NIPS*, 2014. 1
- [24] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 5
- [25] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *ICCV*, 2015. 2
- [26] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015. 2
- [27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 2, 5, 7
- [28] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 1, 2
- [29] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 2, 3, 5, 7
- [30] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2
- [31] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015. 1
- [32] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, 2016. 2, 7
- [33] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. 5
- [34] M. D. Zeiler. Adadelata: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 5

- [35] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. [1](#), [2](#)
- [36] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. [2](#)